IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 3 (Mar. - Apr. 2013), PP 01-05
www.iosrjournals.org

Performance Improvement Techniques for Customized Data Warehouse

Md. Al Mamun and Md. Humayun Kabir
Department of Computer Science and Engineering, Jahangirnagar University, Savar, Dhaka-1342, Bangladesh.

Abstract: *In this paper, we present performance improvement techniques for data retrieval from customized data warehouses for efficient querying and Online Analytical Processing (OLAP) in relation to efficient database and memory management. Different database management techniques, e.g. indexing, partitioning etc. play vital role in efficient memory management. A comparison of data retrieval time for a particular query from a relational database as well as data warehouse database with and without indexing is performed. We show that the application of different database management techniques result faster query execution by reducing data retrieval time. This improved efficiency may increase the efficiency of OLAP operations, which results better data warehouse performance.*

Keywords - *Data Warehouse, Indexing, OLAP, Partitioning, Querying.*

I. Introduction

Data retrieval from relational database requires high access time when it stores millions of records, which can be overcome using indexing [1]. Test data can be generated to populate test databases to test SQL queries [2]. Different database management techniques e.g. partitioning [3], indexing [1, 4, 5] etc. can be employed for faster data access. Customized database application software may be developed using these techniques for faster data processing. In the cases where the number of records in relational database becomes very high, the query processing time becomes very long [6]. Data warehouses [7-9] which store consolidated historic data can be constructed from relational databases. Data warehouse (DW) database store very small number of records for a large number of records in relational database [6]. This reduction of the number of records in data warehouse results in smaller query retrieval time [6]. Moreover, this retrieval time can be further reduced using different indexing techniques.

In this paper, we have studied different techniques to improve performance of data retrieval from relational database as well as DW [8, 9] database for different sizes of data. We have used bitmap indexing (BMI) [10] and data partitioning techniques to improve performance of data retrieval from data warehouse database [6]. We have also shown that data retrieval time for data warehouse is very much lower compared to relational database for similar queries using bitmap indexing [10]. This suggests that data warehouse with bitmap indexing is more suitable for an enterprise for intelligent and faster decision making [3, 6]. The retrieval time rises with the increase in data size.

The paper is organized as follows. Section II presents system architecture, section III describes the data retrieval time measurement technique, section IV presents comparison of data retrieval time for RDBMS with and without indexing, section V presents about data retrieval time for partitioning, section VI presents comparison of data retrieval time for different data sources with variable data sizes, section VII concludes.

II. System Architecture

This section presents the architecture for constructing customized data warehouses from RDBMS tables using data definition language (DDL) of SQL. Data warehouse is populated with data from external sources e.g. relational database system, in the consolidated form using aggregation operation. We have created dimension and fact tables from RDBMS table schemas. Queries are executed on both RDBMS tables and DW dimension and fact tables with and without indexing using Java programs. The data retrieval time are measured and compared. Fig 1 represents the system architecture.

III. Data Retrieval Time

We have designed an algorithm for determining data retrieval time for relational database system and DW database system [6]. We have used the function DBTime () to determine data retrieval time: $t_r = (t_{end} - t_{start})$ in milliseconds (ms) of executing a particular query.

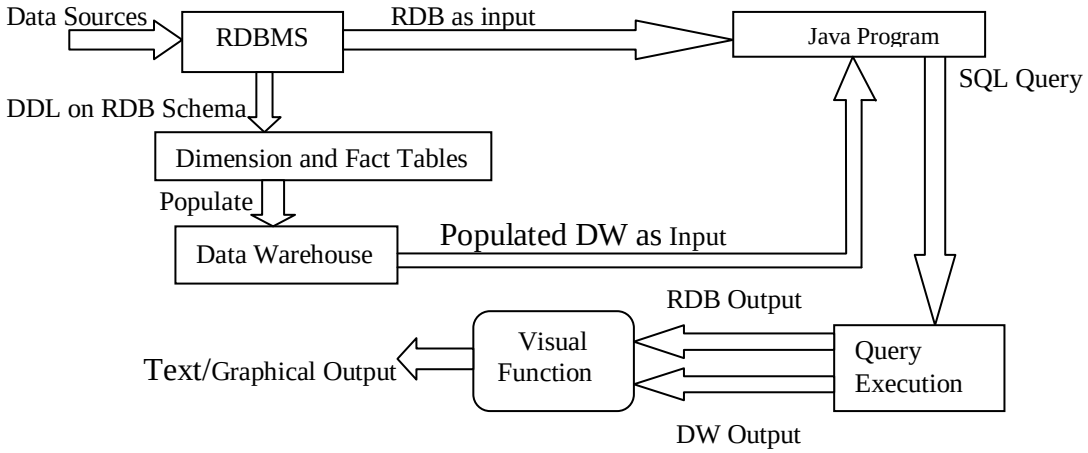


Figure 1: System Architecture

Retrieval time varies with primary memory and processing speed of computer in which we execute our software system. The experiment is done on a laptop with Core i3 processor of 2.53 GHz, RAM 2GB, HD 500GB under windows OS.

IV. Comparison Of Data Retrieval Time For Relational Databases With And Without Indexing

We have applied indexing on students records stored in database relations using Oracle RDBMS. The developed performance improvement system generates very large number of student random data records by varying their CGPAs to populate the RDBMS tables. The developed prototype determines the data retrieval time using data tables created of different data sizes using RDBMS for a particular query without indexing at first as shown in Table 1 [6].

Table 1 Data retrieval time without indexing and with indexing on tables of different data sizes.

Number of Records	**Retrieval Time (ms)**	
Data size in number of records	**Without indexing**	**With indexing**
100000	3324	2819
200000	10671	5457
300000	47727	35340

The database relations are then indexed and the same query is executed on the indexed relations. The query execution times are recorded as shown in Table 1. Fig 2 plots query execution time without and with indexing on different relations of variable data sizes. We notice that the data retrieval time for non-indexed relations is more than that of indexed relations for a particular data size.

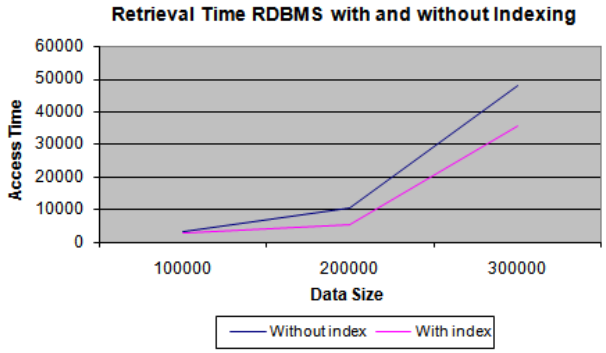


Figure 2: Comparison of data retrieval time without indexing and with indexing for RDBMS.

V. Data Retrieval Time Using Data Partitioning

Data partitioning can speed up the performance of data processing in data retrieval. In case of small physical memory, the large volume of data can be partitioned into smaller segments to load into primary memory. This can help to execute the application program to access data larger than the main memory size successfully. But if the number of partitions is big enough, then data access

may take longer time due to switching between partitions as overhead [3]. Increasing physical memory size, partitioning can be avoided or number of partitions can be reduced resulting faster data access. Partitioning with indexing may cause even more reduction in data retrieval time.

VI. Comparison Of Data Retrieval Time For Different Data Sizes

Consider the data retrieval time shown in Table 2 required to select 209 students of a particular session 2002-2003, who obtained CGPA 4.0 out of 100000 student records by executing a query. We have defined and executed queries to retrieve records of Table 3 from RDBMS, DW without and with bitmap indexing.

Table 2 Data retrieval time (in ms) for RDBMS, Data Warehouse without indexing, and Data Warehouse with BMI

RDBMS (indexed) access time	DW (non-indexed) access time	DW with BMI access time
92	41	28

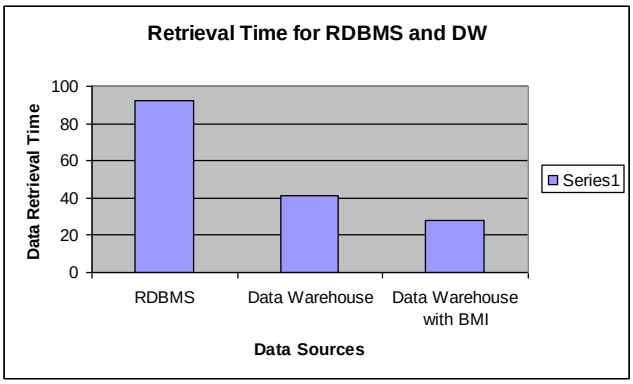


Figure 3: Retrieval time for indexed RDBMS, Data Warehouse without indexing, and Data Warehouse with BMI for selecting 209 students of a session.

Consider the queries to retrieve session, final exam year, CGPA, and the number of the students who obtained CGPA 4 as shown in Table 3. The queries retrieve and count student's records which are stored in RDBMS and DW. Table 4 represents data access time in *ms* for indexed RDBMS, non-indexed DW and DW with bitmap indexing.

TABLE 3 Query Output for students of all sessions with CGPA 4.0

Session	4th Year Final Exam	CGPA	No. of Students
2001-2002	2005	4	503
2002-2003	2006	4	209
2003-2004	2007	4	236
2004-2005	2008	4	265
2005-2006	2009	4	241
2006-2007	2010	4	239
2007-2008	2011	4	261

TABLE 4 Data retrieval time for students of all sessions with CGPA 4.0

Indexed RDBMS	Data Warehouse	Data Warehouse with BMI
555	223	207

Fig 4 plots the data retrieval time shown in Table 4 for different data sources to retrieve the query output shown in Table 3.

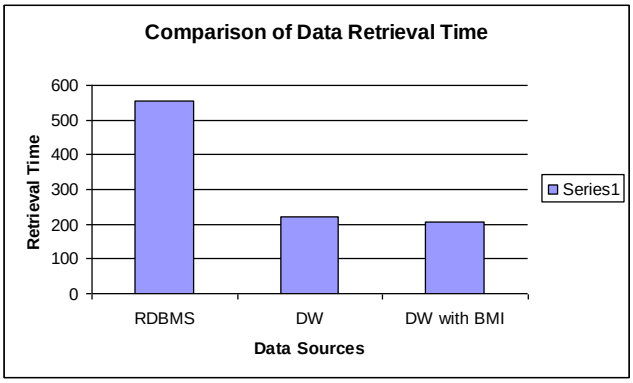


Figure 4: Comparison of data retrieval time for different data sources without and with indexing

Queries operated on databases of RDBMS require the highest time as it stores a large number of raw data records. Table 5 represents the data retrieval time of executing a query on data tables of various sizes containing records of up to 4 millions using RDBMS and the corresponding records in DW. We have measured the execution time of a query for accessing data from an indexed RDBMS database, non-indexed data warehouse database and a DW with bitmap indexing. It is observed that data retrieval from data warehouse with bitmap indexing requires less time compared to that of data warehouse without indexing.

TABLE 5 Retrieval time for CGPA 4.0 students of all sessions from indexed data tables of different sizes stored in RDBMS, DW without and with bitmap indexing

RDBMS	**Data Retrieval Time (ms)**		
No. of Records	**Indexed RDBMS**	**DW**	**DW with BMI**
100000	555	223	207
300000	8964	538	502
500000	13000	769	435
1000000	16259	8569	7520
2000000	42546	17927	17222
4000000	87028	53057	38213

We create two tables Table 6 and Table 7 based on Table 5. Finally, we create another two tables Table 7 and Table 8 for clarification of the data retrieval time for different data sizes of RDBMS database and DW database separately.

TABLE 6 Corresponding Data Sizes of DW database for RDBMS database

Data Size of RDBMS	**Data size in DW**
100000	1954
300000	5879
500000	9880
1000000	19986
2000000	40060
4000000	79803

TABLE 7 Retrieval Time for different Data Sizes of RDBMS

RDBMS Data Size	**RDBMS Retrieval Time**
100000	555
300000	8964
500000	13000
1000000	16259
2000000	42546
4000000	87028

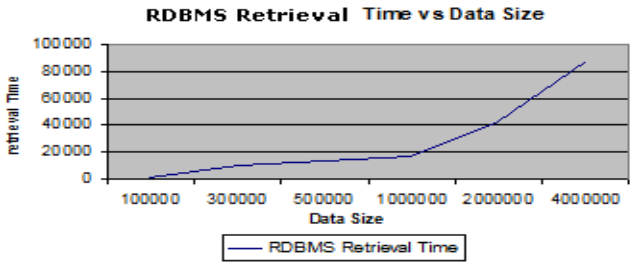
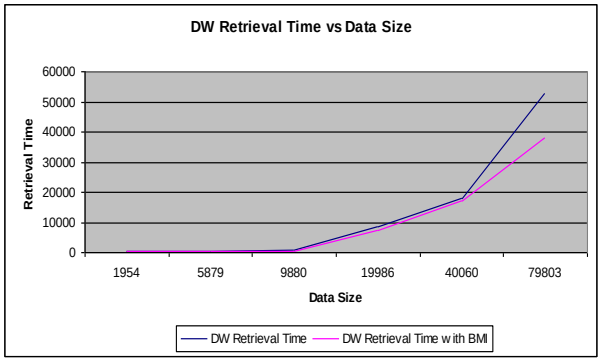


Figure 5: Comparison of data retrieval time for different data Sizes shown in Table 7.

TABLE 8 Retrieval Time for Data Warehouses of different Data Sizes.



Data size of DW	DW Retrieval Time	DW Retrieval Time with BMI
1954	223	207
5879	538	502
9880	769	435
19986	8569	7520
40060	17927	17222
79803	53057	38213

Figure 6: Comparison of data retrieval time for different data Sizes of Data Warehouse with and without BMI shown in Table 8.

Fig. 5 and Fig. 6 explain that data retrieval time in both RDBMS and DW cases increase with the increase in data size significantly.

VII. Conclusion

We have observed that data retrieval from tables stored in RDBMS database is almost exponentially rising. But the increases in data retrieval time for data warehouse with bitmap indexing or without bitmap indexing have small increase with the increase of data size. We have shown that data retrieval time for data warehouse is very much lower compared to relational database for similar queries. This suggests that data warehouse is more suitable for an enterprise for intelligent and faster data access in decision making. This suggests that OLAP system can be developed using DW database and is more suitable than using relational database system for intelligent and efficient decision making with reporting or data analysis.

Acknowledgements

We greatly acknowledge the valuable comments of the faculty members who were present at the thesis examination board.

References

[1] M. Barrena, C. Pachon and E. Jurado, *JISBD2007-04: Neighbors search in holey multidimensional spaces*, IEEE Latin America Transactions, Vol. 6, No. 4, Aug. 2008, pages 332-338.

[2] M. J. Suarez-Cabal, C. de la Riva and J. Tuya, JISBD04-*Populating Test Databases for Testing SQL Queries*, IEEE Latin America Transactions, Vol. 8, No. 2, April 2010, pages 164-171.

[3] Mafruz Zaman Ashrafi, David Taniar, Kate Smith, *ODAM: An Optimized Distributed Association Rule Mining Algorithm*, Monash University, IEEE Distributed Systems Online, IEEE Computer Society, Vol. 5, No. 3; March 2004.

[4] Bernd Reiner, Karl Hahn. *Optimized Management of Large-Scale Data Sets Stored on Tertiary Storage System*, IEEE Distributed Systems Online, IEEE Computer Society, Vol. 5, No. 5; May 2004.

[5] S. Repp, A. Gross and C. Meinel, *Browsing within Lecture Videos based on the chain index of speech transcription*, IEEE transactions on learning technologies, Vol. 1, Issue 3, 2008, pages 145-156.

[6] M. Al Mamun. *Data Warehouse Performance Analysis for Online Analytical Processing*, A Thesis Draft submitted for predefense of MS, Department of Computer Science and Engineering, Jahangirnagar University, Savar, Dhaka-1342.

[7] V. Nebot and R. Berlanga, JISBD02-*Populating Data Warehouses with Semantic Data*, IEEE Latin America Transactions, Vol. 8, No. 2, April 2010, pages 150-157.

[8] J.-N. Mazon and J. Trujillo, *JISBD2007-02: Model-driven reverse engineering for data warehouse design*, IEEE Latin America Transactions, Vol. 6, No. 4, Aug. 2008, pages 317-323.

[9] E. Soler, J. Trujillo, E. Fernandez-Medina, and M. Piattini, *JISBD2007-07: An extension of the relational metamodel of CWM to represent secure data warehouse at the logical level*, IEEE Latin America Transactions, Vol. 6, No. 4, Aug. 2008, pages 355-362.

[10] Morteza Zaker, Somnuk Phon-Amnuaisuk and Su-Cheng Haw, *An Adequate Design for Large Datawarehouse systems: Bitmap indexes versus B-tree index*, International Journal of Computers and Communications, Vol. 2, Issue 2, 2008.